

MODÜL 10

MODÜLER PROGRAMLAMA



Şekil 10.1: Bölümle ilgili örnek uygulamalara karekoddan ulaşabilirsiniz.

Modülerin bir diğer anlamı da birbiriyle uyumlu olan ve önceden hazırlanmış parçalardır. Bu parçalar bir araya gelerek uyumlu bir şekilde bir bütünü oluşturmaktadır. Python’da aslında her bir dosya modüldür. Modüler içinde fonksiyonları sınıfları ve objeleri bulundurur. Bu modülleri projemize dâhil edersek içerisinde bulunan fonksiyonları, sınıfları ve objeleri kullanabiliriz. Python modüler yapıyı destekleyen bir programlama dilidir. Python birçok modül içerdiği gibi, kullanıcı kendisi de modül yazıp kullanabilir.

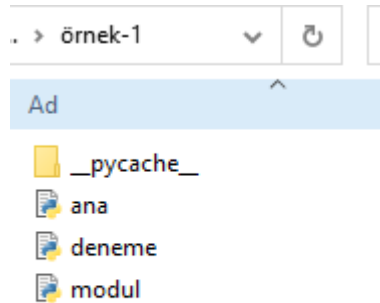
Programın önceden yazılmış parçaları, bütüne istenildiği zaman dâhil edilerek çalıştırabilir. Böylelikle programcının çalıştığı ortam sadece kendi işine yaradığı modülleri kullanacağından gereksiz yere bellek tüketimi olmayacaktır. Python birçok modül içerdiği gibi, kullanıcı kendisi de modül yazabilir. Python modülleri kütüphane olarak da adlandırılır.

Modül Kullanmanın Avantajları:

- Modüller, kod tekrarını önler.
- Modüller, projemizin daha okunabilir olmasını sağlar.
- Modüller yapıda projeyi değiştirmek, güncellemek, yeni modüller eklemek daha basittir.
- Modüller yapıda projede bakım kolaydır.
- Modüller yapıda aynı projede birden fazla kişinin çalışmasına imkân sağlar.

10.1 Modül Yazma ve Çağırma

Modül oluşturmak ve projeye dâhil etmek için bir Python dosyası oluşturulur. Çünkü Python dosyası da bir modüldür. “modul.py” adında bir Python dosyası oluşturup kaydedilir. Örnek 1, örnek 2 adında devam edecektir. Bunların her biri ayrı klasörlerdir. Bu klasörlerin içine Python dosyaları oluşturulur. Örnek 1’in klasör yapısı Şekil 10.2’deki gibi düzenlenir. __pycache__ klasörünü Python’un kendisi modül yazınca otomatik atmaktadır.



Şekil 10.2: Klasör yapısı

Örnek

1

```
# modul.py
def cagir():
    print("Merhaba Öğretmen Arkadaşım")
```

Oluşturduğunuz modülü çağırmak için;

Import modül adı şeklinde çağrılır.

Başka bir Python dosyası oluşturulur. Buna da ana.py adı verilir. Bu dosya çalıştırıldığında ekran çıktısı aşağıdaki gibi olacaktır. Modülün içinde neler olduğunu incelediğimizde “dir” fonksiyonu ile modülün içindeki metotları görmekteyiz. Listenin son elemanı ise oluşturulan cagir fonksiyonudur.

```
#ana.py
import modul
print(dir(modul))

['__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'cagir']
```

Cagir fonsiyonunu kullanmak için **deneme.py** adında bir Python dosyası oluşturunuz. Bu Python dosyasının içine aşağıdaki kodları yazınız.

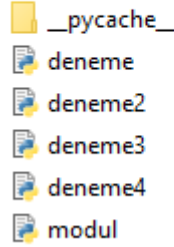
```
#deneme.py
import modul #modülü çağırdık
modul.cagir()#modül içindeki fonksiyonu çalıştırdık
Merhaba Öğretmen Arkadaşım
```

Bir modül boş bir dosyadan oluşabileceği gibi, çeşitli değişkenlerin bulunduğu bazı işlevlerin yazıldığı, bir veya birden fazla sınıfın bulunduğu karmaşık yapıya da sahip olabilir. Örnek 2’de değişkenler tanımlanıp, bu değişkenler modül yardımıyla çağırılmıştır. Modülünüzü oluşturup **modul.py** dosyası olarak kaydedebilirsiniz.

MODÜL 10

örnekler > örnek-2

Ad



Şekil 10.3: Örnek 2 klasör yapısı

Örnek

2

```
#modul.py
ad = 'Cemal'
soyad = 'Türk'
yas=60
meslek='Kaptan'
```

Artık yazılan modül içerisindeki değişkenlere ulaşılabilir. Eğer bir modülü bu şekilde adını kullanıp eklenirse modül içerisindeki nesnelere <modüladı>.<nesneadı> şeklinde ulaşılabilme imkanı sağlar. Şimdi modülü çağırmak için deneme.py adlı Python dosyası oluşturulur. Bu dosyanın içine aşağıdaki kodları yazılır. Görüldüğü üzere modülünüzü modül adı, değişken adı şeklinde çağırarak **deneme.py** adlı Python dosyasında çalıştırabilirsiniz.

```
#deneme.py
import modul
print('Adı      :',modul.ad)
print('Soyadı   :',modul.soyad)
print('Yaşı     :',modul.yas)
print('Mesleği  :',modul.meslek)
Adı      :Cemal
Soyadı   :Türk
Yaşı     :60
Mesleği  :Kaptan
```

Program içinde sadece nesneleri sadece ismi ile çağırmak istenebilir. Bundan dolayı modülde sadece o nesneyi çağırmak gerekebilir.

Kullanımı: from <Modül adı> import <İçericek Nesne>

Örnek 2 klasörünün içinde **deneme2.py** dosyası oluşturulur. Bu dosyanın içerisine aşağıdaki kodları yazalım. Sadece kişiye ait meslek bilgisini çekmek istediğimizi farz ediyoruz. Ekrana böylelikle modül sayfamızdaki sadece meslek nesnesinin değeri gelmiş olur. Bu şekilde modül içindeki sadece tek nesneye değil de diğer nesnelere ulaşmak istenildiğinde ise hata verecektir.

```
#deneme2.py
from modul import meslek
print('Mesleği:',meslek)
Mesleği:Kaptan
```

Tüm nesneler kullanılmak istendiğinde ise; **from <modül> import*** şeklinde tanımlama gerekmektedir. **deneme3.py** adında bir Python dosyası oluşturularak, içine aşağıdaki kodları yazınız. Bu şekilde kullanımında **modul.py** dosyasındaki nesneleri ekrana yazdırmak için tekrar modül ismini yazmamız gerekmektedir. Sadece nesne ismini yazarak da çağırabilirsiniz.

```
#deneme3.py
from modul import*
print('Adı      :',ad)
print('Soyadı   :',soyad)
print('Yaşı     :',yas)
print('Mesleği  :',meslek)
Adı      : Cemal
Soyadı   : Türk
Yaşı     : 60
Mesleği  : Kaptan
```

Tüm nesneleri kullanmak istiyoruz ve modülümüze takma isim verilerek de çağırma işlemi gerçekleştirilir.

Kullanımı: import modul_adi as takma_adi

MODÜL 10

Örnek 2 klasörümüzün içinde deneme4.py adlı bir Python dosyası oluşturarak içine aşağıdaki kodları yazıyoruz. Böylelikle modülümüzü takma ad ile birlikte çağırmış oluyoruz.

```
#deneme4.py
import modul as mod
print('Adı      :',mod.ad)
print('Soyadı   :',mod.soyad)
print('Yaşı     :',mod.yas)
print('Mesleği: ',mod.meslek)
Adı      : Cemal
Soyadı   : Türk
Yaşı     : 60
Mesleği: Kaptan
```

10.2. Hazır Modülleri Kullanımı

Python’da herhangi bir hazır modülü kullanabilmek için öncelikle o modülü içe aktarılması gerekmektedir. İçe aktarmak bir modül içindeki fonksiyon ve niteliklerin başka bir program (veya ortam) içinden kullanılabilir hale getirilmesi anlamını taşımaktadır. Hazır modülleri kullanmak için modül ismi ile import edilerek çağrılmaktadır. Modül içindeki nesneler ise adlarıyla birlikte kullanılabilir.

Kullanımı: import hazır modül adı

Örnek 3’te math modülünü projeye dâhil ediniz. Modüle ait özellikleri listeleyiniz.

Örnek

3

```
import math
print(dir(math))
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh',
'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh',
'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial', 'floor', 'fmod',
'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf',
'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi',
'pow', 'radians', 'remainder', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']
```

10.3. Random ve Math Kütüphaneleri

Örnek 3'teki math modülüne ait nesneleri adlarını kullanarak projeye ekleyebilirsiniz.

Math Kütüphaneleri

Math.pi, bize pi sayısını ve **math.pow** ise sayının üssünü alarak değerini vermektedir. Bu değerleri kullanarak dairenin alanını Örnek 4'te bulmaya çalışınız. Örnekte önce math modülümüzü çağırdık. Yarıcap değişkenine 4 değerini atayalım. Dairenin alanını math.pi ve math.pow özelliklerini kullanarak alan değişkenine atayarak, ekrana yazdırılmıştır.

Örnek**4**

```
import math
yaricap=4
alan=math.pi*(math.pow(yaricap,2))
print(alan)
50.26548245743669
```

Örnek 5'e bakıldığında, math modül içerisindeki cos(), factorial(), pow(), sqrt() gibi fonksiyonları çağırarak çeşitli işlemler yapılmıştır.

Örnek**5**

```
import math
ustal=math.pow(5,2)
fak=math.factorial(4)
cosinus=math.cos(120)
karekok=math.sqrt(81)
print(ustal)
print(fak)
print(cosinus)
print(karekok)
25.0
24
0.8141809705265618
9.0
```

Random Kütüphanesi, Python’da rastgele sayı üretilmesini sağlamaktadır.

Örnek 6’da, random kütüphanesi çağırarak, bu kütüphane içinde random modülüne ait özellikleri listeleyiniz.

Örnek**6**

```
import random
print(dir(random))

['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONST',
'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Sequence', '_Set',
'_all_', '_builtins_', '_cached_', '_doc_', '_file_', '_loader_', '_name_',
'_package_', '_spec_', '_acos', '_bisect', '_ceil', '_cos', '_e',
'_exp', '_inst', '_itertools', '_log', '_os', '_pi', '_random', '_sha512', '_sin',
'_sqrt', '_test', '_test_generator', '_urandom', '_warn', 'betavariate',
'choice', 'choices', 'expovariate', 'gammavariate', 'gauss', 'getrandbits',
'getstate', 'lognormvariate', 'normalvariate', 'paretovariate', 'randint', 'random',
'randrange', 'sample', 'seed', 'setstate', 'shuffle', 'triangular', 'uniform',
'vonmisesvariate', 'weibullvariate']
```

Örnek 7’de Random kütüphanesini kullanarak içindeki nesneler yardımıyla program yazınız. Modülün bu metodu kullanıldığı zaman 0 ile 1 arasında rastgele bir sayı üretilir. Üretilen bu sayı 0 olabilirken 1 olamaz; yani $[0, 1)$ aralığında oluşmaktadır.

Örnek**7**

```
import random
a=random.random()
print(a)
0.2980492500422903
```

Eğer integer tipinde bir sayı istenirse bu durumda randint metodu kullanmak gerekmektedir. Bu metot kullanılırken başlangıç ve bitiş aralık değerleri verilir; ancak bu durumda bitiş değeri de rastgele sayı olarak tutabilir.

Kullanımı: [başlangıç, bitiş] aralığı kullanılır.

Örnek 8’de Random modülü ile 0 dâhil değil 10 dahil olmak üzere sayı üretmesi sağlanır. Rastgele oluşturulan sayıyı değişkene aktararak, ekrana yazdırınız.

Örnek**8**

```
import random
sayi=random.randint(1, 10)
print(sayi)
3
```

random.choice, ile listeden rastgele bir öge seçilmesine izin vermektedir. Örnek 9’da cicekler adında bir liste tanımlayınız ve elemanlarını ekleyiniz. Liste içerisinde rastgele 3 adet eleman seçerek, ekrana yazdırılmıştır.

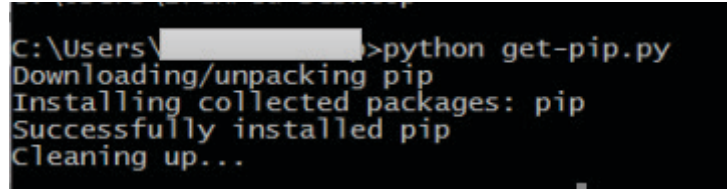
Örnek**9**

```
import random
cicekler = ['gül', 'karanfil', 'papatya','begonya','menekşe']
print(random.choice(cicekler))#1. rastgele seçim
print(random.choice(cicekler))#2. rastgele seçim
print(random.choice(cicekler))#3. rastgele seçim
gül
menekşe
menekşe
```

10.4. Pip Paket Yükleyici Kullanımı

Pip bir paket yöneticisidir. Python kütüphane ve modüllerini kurmanızı sağlayan yardımcı bir programdır. Python’da standart paketlerin dışında bir de üçüncü parti modülleri vardır. Bunlar Python geliştiricileri haricindeki kişilerce yazılıp kullanımımıza sunulmuş araçlardır. Bu paketler, standart paketlerin aksine dilin bir parçası olmadığından, bu paketleri kullanabilmek için, pip yardımıyla kurmanız gerekmektedir. Bu üçüncü parti birçok modüllere <https://pypi.python.org/pypi> adresinden adını öğrenerek, paketleri yüklenebilir. Pip paket yükleyicisini kullanabilmek amacıyla öncelikle pip paket yükleyicisinin yüklenmesi gerekmektedir. Pip yükleyicisinin kurulumunu için python yorumlayıcısından yapabilmekteyiz.

Öncelikle <https://pip.pypa.io/en/latest/installing.html> adresine giderek adreste gösterilmiş olan “get-pip.py” dosyasını bilgisayara indirilmesi gerekmektedir. İşletim sistemine bağlı olarak komut satırını açıp python yorumlayıcısını çalıştırarak Python yorumlayıcısına aşağıdaki komut yazılarak pip yükleyicisinin kurulumu gerçekleştirilir.



```
C:\Users\>python get-pip.py
Downloading/unpacking pip
Installing collected packages: pip
Successfully installed pip
Cleaning up...
```

Şekil 10.4: Pip Paket yükleyici ekranı

Pip ile Paket Nasıl Yüklenir ve Kaldırılır?

Pip ile paket kurmak için cmd açılmalı ve cmd ekranına şu kod yazılmalıdır:

```
pip install paket_adı          # paketi kurmak için
pip uninstall paket_adı       # kurulu paketi kaldırmak için
pip install PaketAdi==1.0.4    # istenilen versiyonu kurar
$ pip install 'PaketAdi>=1.0.4' # alt limit ile verilen versiyonu kurar
```

install	: Yeni bir paket yükler.
Uninstall	: Varolan bir paketi siler.
Freze	: Yüklü tüm paketleri requirements formatında listesini çıktıya verir.
List	: Yüklü tüm paketleri normal listesini çıktıya verir.

Show	: Yüklü paketler hakkında bilgi verir.
Search	: Paketler içinde arama yapar.
Wheel	: Paketler içinde requirements için bir arşiv yapar.
Upgrade	: Kurulu bir paketi güncellemek için kullanılır.

Python2’de pipi kullanmak için pip2 ve Python3’te kullanmak için de pip3 komutu kullanılır. Cmd ekranı kullanılarak Şekil 10.5’te görüldüğü üzere “face_recognition” paketini yükleyebiliriz. Kaldırmak içinde aynı şekilde “uninstall” komutu kullanılır.

```
C:\ProgramData\Anaconda3\Scripts>pip install face_recognition
```

Şekil 10.5: face_recognition paketini yükleme ekranı

Örnek

10

Pip ile django paketini yükleyip, listele işlemini yaparak, yüklü paketler hakkında bilgi alalım.

```
pip install django
Collecting django
  Downloading Django-3.0.5-py3-none-any.whl (7.5 MB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.3.1-py2.py3-none-any.whl (40 kB)
Collecting asgiref~=3.2
  Downloading asgiref-3.2.7-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: pytz in c:\programdata\anaconda3\lib\site-packages (from django) (2019.3)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.2.7 django-3.0.5 sqlparse-0.3.1
Note: you may need to restart the kernel to use updated packages.
```

MODÜL 10

Örnek

11

pip install django ile kurduğunuz paketi yükseltmeye çalışabilirsiniz.

```
pip install django --upgrade
Requirement already up-to-date: django in c:\programdata\anaconda3\lib\site-packages (3.0.5)
Requirement already satisfied, skipping upgrade: pytz in c:\programdata\anaconda3\lib\site-packages (from django) (2019.3)
Requirement already satisfied, skipping upgrade: asgiref~=3.2 in c:\programdata\anaconda3\lib\site-packages (from django) (3.2.7)
Requirement already satisfied, skipping upgrade: sqlparse>=0.2.2 in c:\programdata\anaconda3\lib\site-packages (from django) (0.3.1)
Note: you may need to restart the kernel to use updated packages.
```

Örnek

12

Yüklediğiniz django paketini kaldırabilirsiniz.

```
pip uninstall django
```

10.5. Time Modülü

Time modülü zaman değerlerini düzenlemekle ilgili birçok görevi yerine getirebilmektedir. Zamanı göstermek için iki standart bulunmaktadır. Birincisi zamanı Epoch'tan itibaren saniye olarak vermektir. Epoch Unix zaman başlangıcı olarak alınır ve takvime göre 1 Ocak 1970'e denk gelmektedir. Eğer Epoch türünden şimdiki zaman saniye biçiminde alınmak istenirse şu yöntemi kullanabilirsiniz.

Örnek

13

```
import time
print (time.time())
1587590353.7686868
```

Ekran çıktısı bize 1 Ocak 1970'ten itibaren kaç saniye geçtiğini vermektedir. Elde edilen değeri **gmtime()** fonksiyonu kullanarak okunabilir tarih formatına çevrilebilir. Örnek 14'te saniye cinsinden elde edilen zaman değeri, okunabilir tarih formatına çevrilmiştir.

Örnek

14

```
import time
print (time.gmtime(time.time()))
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=22, tm_hour=21, tm_min=44, tm_
sec=9, tm_wday=2, tm_yday=113, tm_isdst=0)
```

Localtime, zaman bilgilerini sıralı bir tüp şeklinde vermektedir. Örnek 15'te time modülünde localtime nesnesini kullanarak zaman bilgilerini sıralı bir şekilde işlemi yapmaktadır.

Örnek

15

```
import time
print (time.localtime())
time.struct_time(tm_year=2020, tm_mon=4, tm_mday=23, tm_hour=0, tm_min=25, tm_
sec=31, tm_wday=3, tm_yday=114, tm_isdst=0)
```

ctime fonksiyonu, içinde bulunulan zaman bilgilerini vermektedir. Örnek 16'da ctime fonksiyonu ile güncel tarih bilgilerini yazmaktadır.

Örnek

16

```
import time
print (time.ctime())
Thu Apr 23 00:36:07 2020
```

strftime() fonksiyonu ile kendimize ait zaman cümlesi oluşturabilirsiniz. Bu zaman cümlesinin belirlediğimiz duruma göre ekran çıktısının verilmesini sağlar. Tablo 1'de strftime() fonksiyonuna ait yönergeler verilmektedir. Bu yönergeler ister tek başına istersek yönergeleri birleştirerek de kullanabiliriz.

Tablo 1. strftime() fonksiyonu yönergeleri ve anlamları

Yönerge	Anlamı
%a	Kısaltılmış gün adı
%A	Gün adı
%b	Ayın kısaltılmış adı
%B	Ayın adı
%c	Tam tarih ve saat
%d	Ayın günü (01-31)
%H	Saat (00-24)
%I	Saat (01-12)
%j	Gün (01-366)
%m	Ay (00-12)
%M	Dakika (00-59)
%p	Öğleden önce (ÖÖ), öğleden sonra (ÖS)
%S	Saniye (00-59)
%U	Yılın kaçınıcı haftası (00-53)
%w	Haftanın kaçınıcı günü (0-6)
%y	Yılın son iki hanesi (15)
%d	Ayın günü (örnek: Nisan için 13)
%Y	Yıl

Örnek 17’de gün ay yıl yönergelerini kullanarak yan yana yazımı verilmiştir. Bu şekilde yönergeleri ekleyerek, güncel tarih bilgilerini strftime() fonksiyonu ile alabiliriz.

Örnek

17

```
import time
print (time.strftime("%d/%m/%Y"))
23/04/2020
```

sleep() Fonksiyonu, programın belirlenen süre boyunca durdurulmasına olanak sağlar. Aldığı argüman saniye cinsindendir. Örnek 18'deki kodlar çalıştırıldığında 10 saniye program duraklar.

Örnek

18

```
import time
time.sleep(10)
```

Örnek 19'da ise bugünün tarihini saniyede içinde olacak şekilde verilmiştir. Ekran çıktısı olarak başlangıç tarihi ile bitiş tarihi arasında 5 saniye olduğu görülmektedir.

Örnek

19

```
import time
print ("Başlangıç : %s" % time.ctime())
time.sleep( 5 )
print ("Bitiş : %s" % time.ctime())
Başlangıç : Thu Apr 23 01:19:27 2020
Bitiş : Thu Apr 23 01:19:32 2020
```

10.6. Bölüm Sonu Örnekleri

1. 0-20'ye kadar sayıları döngü yapısı ile ekrana yazdırırken her bir sayı arasına yarım saniye duraklamalar koyarak programı yapınız.
2. Os modülünde name fonksiyonu işletim sistemi hakkında bilgi vermektedir. İşletim sisteminiz hakkında bilgi almak için gerekli olan kodları yazınız.
3. İki sayının toplamını yaptıran programı fonksiyon ve modül yazma kullanarak yapınız.

Cevaplar

```
1. import time
   for i in range(1,21):
       time.sleep(0.5)
       print(i)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

```
2. import os
   print (os.name)
Nt
```

MODÜL 10

3. #modul.py dosyası;

```
def toplama(sayi1,sayi2):  
    toplam=int(sayi1)+int(sayi2)  
    return toplam
```

#deneme.py dosyası;

```
import modul  
a = input("birinci sayıyı giriniz:")  
b = input("ikinci sayıyı giriniz:")  
sonuc = int(modul.toplama(a, b))  
print (sonuc)  
birinci sayıyı giriniz:6  
ikinci sayıyı giriniz:7  
13
```